SQL Assignments

28. Write a short essay talking about your understanding of transactions, locks and isolation levels.

The primary benefit to use transaction is data integrity. Therefore, transaction follows ACID rules which are atomicity, consistency, isolation, and durability. Atomicity ensures that a transaction either success or fail with no between. Consistency means that transactions behave the way we are expected. Isolation helps when multiple users reading or writing from the same table all at once, their transactions won’t interfere with or affect one another. Durability ensures that changes to your data made by successfully executed transactions will be saved, even in the event of system failure. Because of concurrency issues that may happen in some scenario, we have different levels of isolation levels from least restrict to most restrict. For example: snapshot, read uncommitted, read committed (system default), repeatable read, and serializable. Locks are used to achieve different levels of isolation. One common lock is dead lock. It happens when two processes need both two tables to complete its process, but one process locked one table and another process locked another table. Then they have to wait for each other to release the table, therefore, neither of them can finish because the other table has been locked.

1. Graphical user interface, text, application

   Description automatically generatedWrite a short essay, plus screenshots talking about performance tuning in SQL Server. Must include Tuning Advisor, Extended Events, DMV, Logs and Execution Plan.

Diagram

Description automatically generatedFrom above execution plan we see that the performance of query 1 and 2 are good, the total cost time is less than one second.

Performance of query 3 is over 6mins. At start, it did not cost much even we have two table scans because the tables are relatively small. Then it did a hash match join and clustered index scan quickly. The next join and aggregation cost almost the whole time to compute because it received a lot a data.

Graphical user interface, text, application, email

Description automatically generatedBecause I am using SQL server express, I am not able to use Tuning Advisor.

This is a live data from extended event generated by simple query. From above information, we can see that this query is efficient.

Group assignments:

1. Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called “Adventure works”! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.
2. Database Design: OLTP db design request for EMS business: when people call 911 for medical emergency, 911 will dispatch UNITs to the given address. A UNIT means a crew on an apparatus (Fire Engine, Ambulance, Medic Ambulance, Helicopter, EMS supervisor). A crew member would have a medical level (EMR, EMT, A-EMT, Medic). All the treatments provided on scene are free. If the patient needs to be transported, that’s where the bill comes in. A bill consists of Units dispatched (Fire Engine and EMS Supervisor are free), crew members provided care (EMRs and EMTs are free), Transported miles from the scene to the hospital (Helicopters have a much higher rate, as you can image) and tax (Tax rate is 6%). Bill should be sent to the patient insurance company first. If there is a deductible, we send the unpaid bill to the patient only. Don’t forget about patient information, medical nature and bill paying status.

Entity table: Crews, Patient, Apparatus, Events.

Look up table: Location, Bill, Insurances

Junction table: Crew&Event, Apparatus&Event, Crew&Apparatus, Bill&Apparatus, Bill&Crew

Crew table with columns: CrewID int pk, CrewName varchar, CrewType Varchar, EMR binary, EMT binary, A-EMT binary, Medic binary, AdditionalInfo Varchar, Crew&EventId int fk, Crew&ApparatusId int fk, Bill&CrewId int fk

Patient table with columns: PatientId int pk, PatientName Varchar, MedicalNature Varchar, BillPayingStatus binary, AdditionalInfo Varchar, InsuranceId int fk, InsurancePay Decimal, locationId int fk, Deductible decimal

Apparatus table with columns: ApparatusId int pk, Name Varchar, Type Varchar, Apparatus&EventId int fk, Bill&ApparatusId int fk, ChargeRate Decimal

Events table with columns: EventId int pk, PatientId int fk, LocationId fk, BillId int fk,

EventDetail Varchar, Crew&EventId int fk, Apparatus&EventId int fk, Crew&ApparatusId int fk, MilesToHospital Int

Location table with columns: LocationId int pk, Address1 Varchar, Address2 Varchar, City varchar, County Varchar, State varchar, AdditionalInfo Varchar.

Bill table with columns: BillId int pk, TaxRate decimal, Bill&ApparatusId int fk, InsuranceId int fk, AdditionalInfo Varchar, Bill&ApparatusId int fk, Bill&CrewId int fk, BillTotal int

Insurances table with columns: InsuranceId int pk, Name Varchar, AdditionalInfo Varchar

Crew&Event table with columns: Crew&EventId int pk, CrewId int fk, EventId int fk

Apparatus&Event table with columns: Apparatus&EventId int pk, ApparatusId int fk, EventId int fk

Crew&Apparatus table with columns: Crew&ApparatusId int pk, CrewId int fk, ApparatusId int fk

Bill&Apparatus table with columns: Bill&ApparatusId int pk, BillId int fk, ApparatusId int fk

Bill&Crew table with columns: Bill&CrewId int pk, BillId int fk, CrewId int fk

1. Remember the discussion about those two databases from the class, also remember, those data models are not perfect. You can always add new columns (but not alter or drop columns) to any tables. Suggesting adding Ingested DateTime and Surrogate Key columns. Study the Wide World Importers DW. Think the integration schema is the ODS. Come up with a TSQL Stored Procedure driven solution to move the data from WWI database to ODS, and then from the ODS to the fact tables and dimension tables. By the way, WWI DW is a galaxy schema db. Requirements:
   1. Luckly, we only start with 1 fact: Purchase. Other facts can be ignored for now.
   2. Add a new dimension: Country of Manufacture. It should be given on top of Stock Items.
   3. Write script(s) and stored procedure(s) for the entire ETL from WWI db to DW.